2’s Complements:

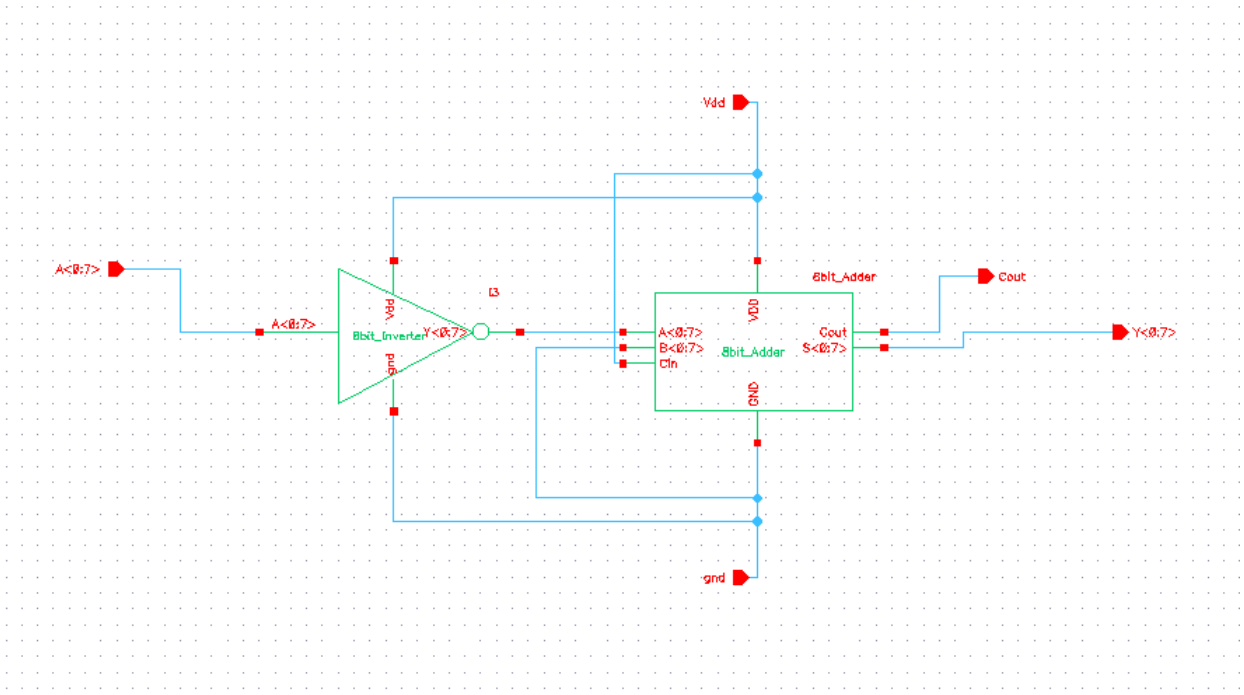
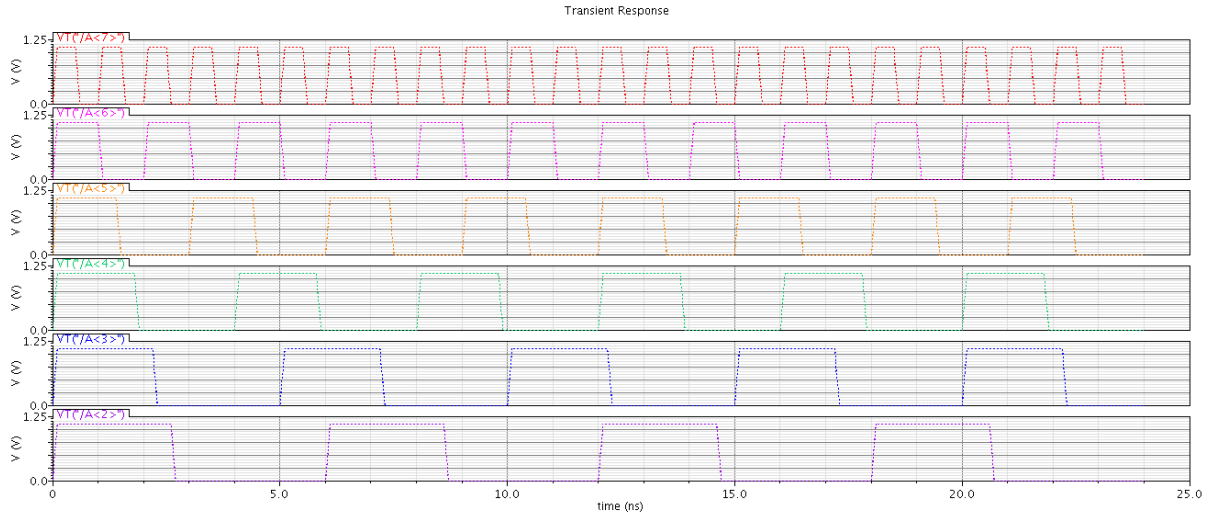


Figure 1: This is the 8 bit two’s complements we created for the ALU.



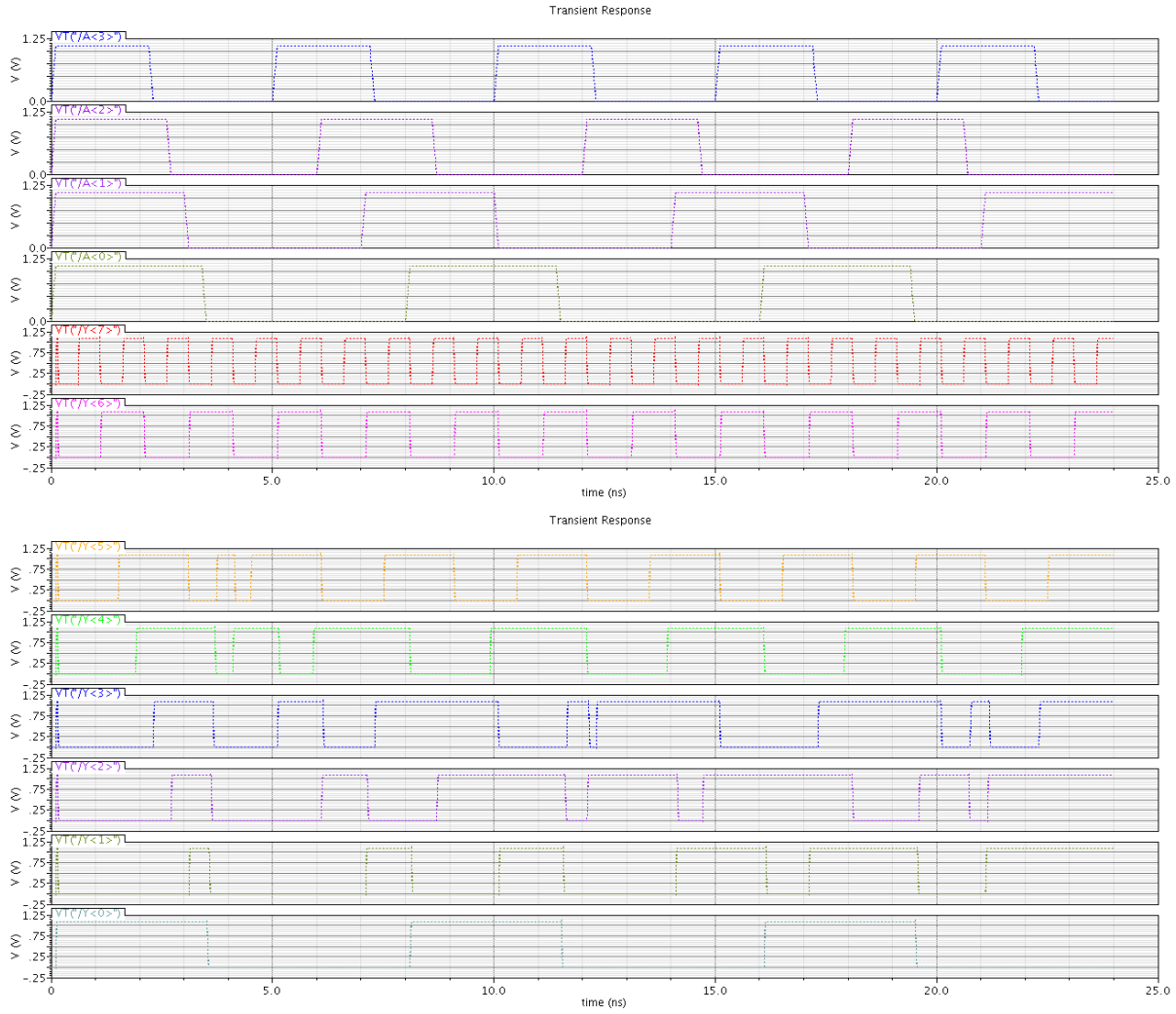


Figure 2,3,4: Are the simulations that correspond for the 8 bit two's complement for each bit. We used these simulations to calculate the worst case delay and the scenario that corresponds to it.

Adder:

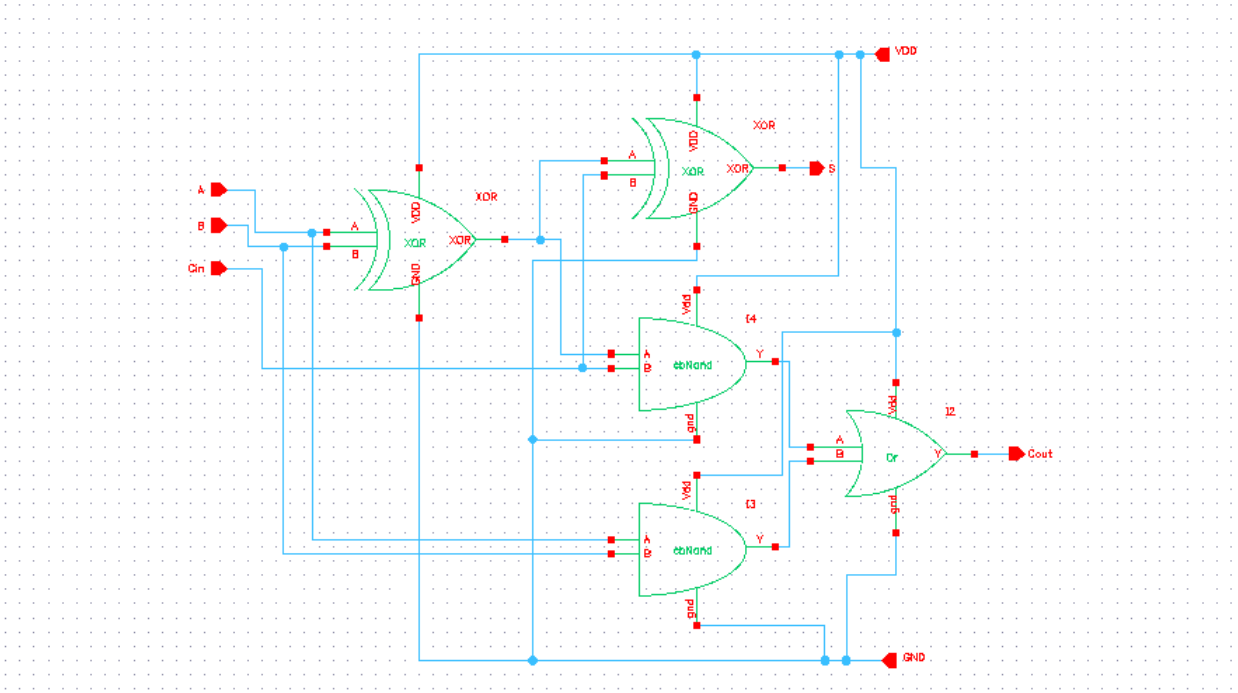


Figure 5: This is the basic full adder that we created that we would use to build the 8 bit adder and the two's complements.

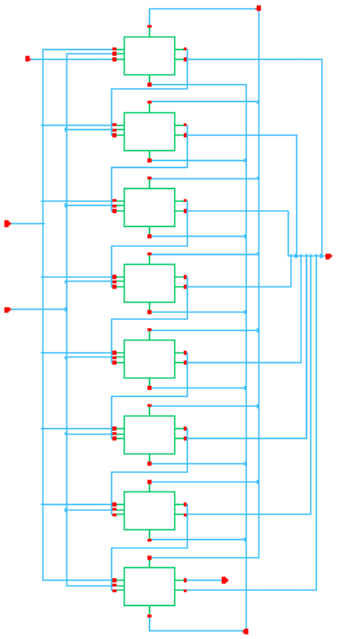


Figure6: The 8 bit adder that we created using the basic full adder we previously created.

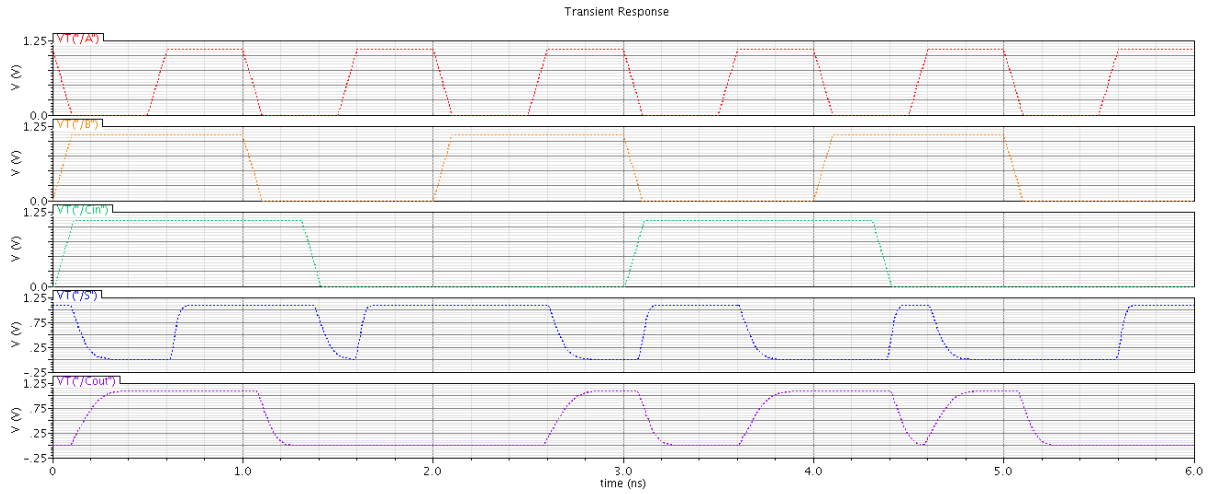


Figure 7: Simulation for the basic full adder we created. We used these simulations to calculate the worst case delay and the scenario that corresponds to it.

And:

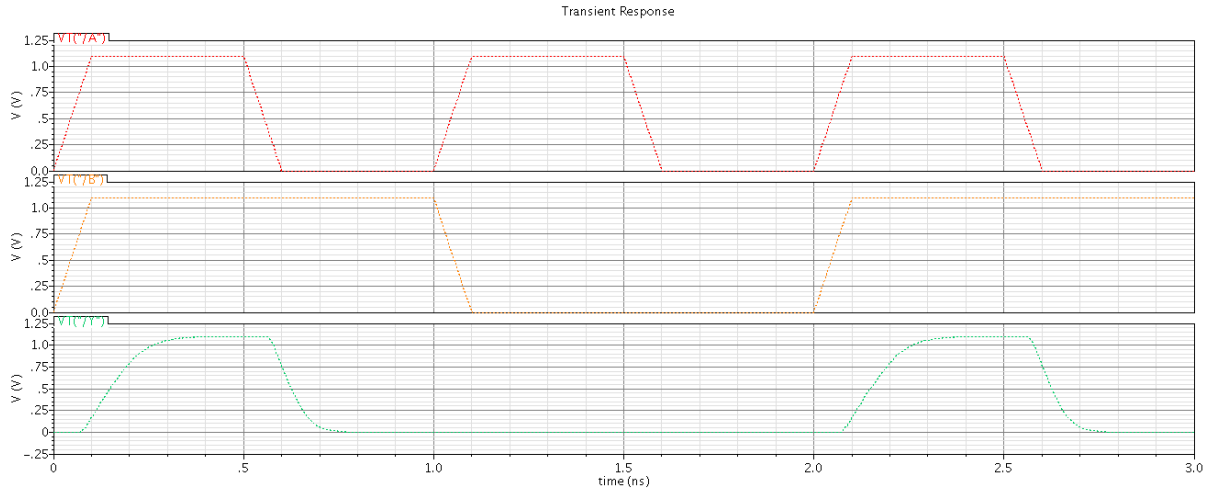


Figure 8: Simulation for the basic And component we created that we used to calculate the worst case delay and the scenario that corresponds to it.

Or:

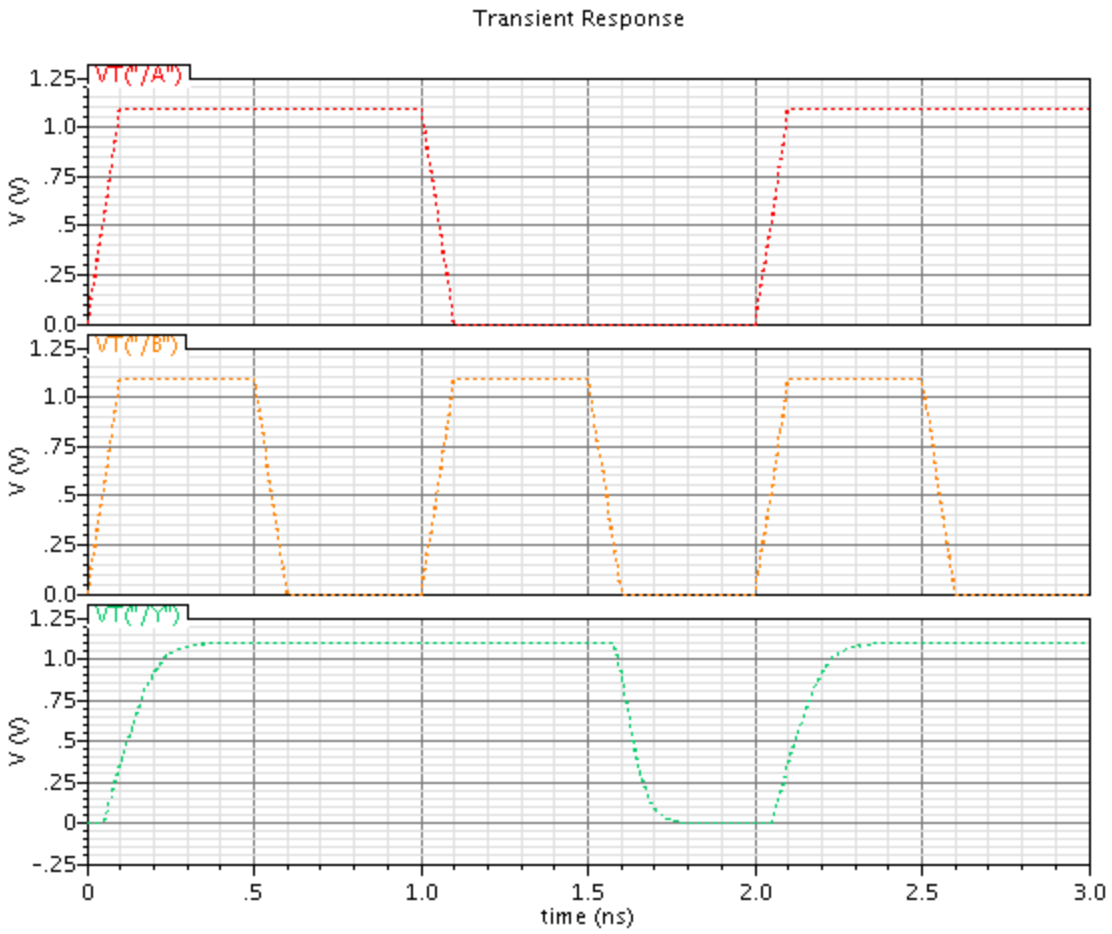


Figure 9: Simulation for the basic Or component we created that we used to calculate the worst case delay and the scenario that corresponds to it.

PassA:

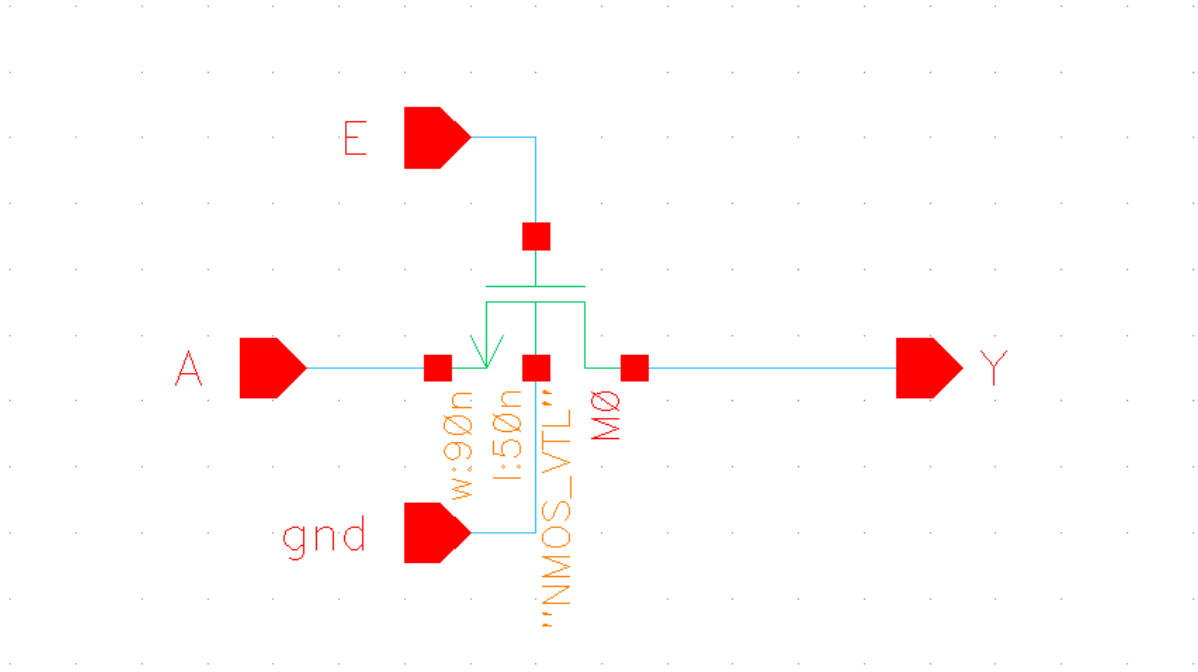


Figure 10:This the schematic of the basic pass A component.

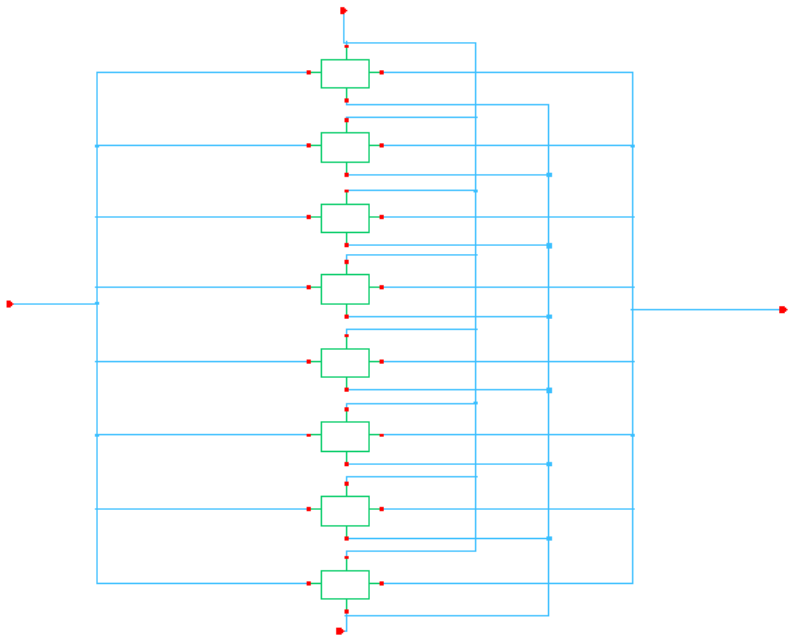


Figure 11: This is the schematic of the 8 bit pass A component we created using the basic pass A component.

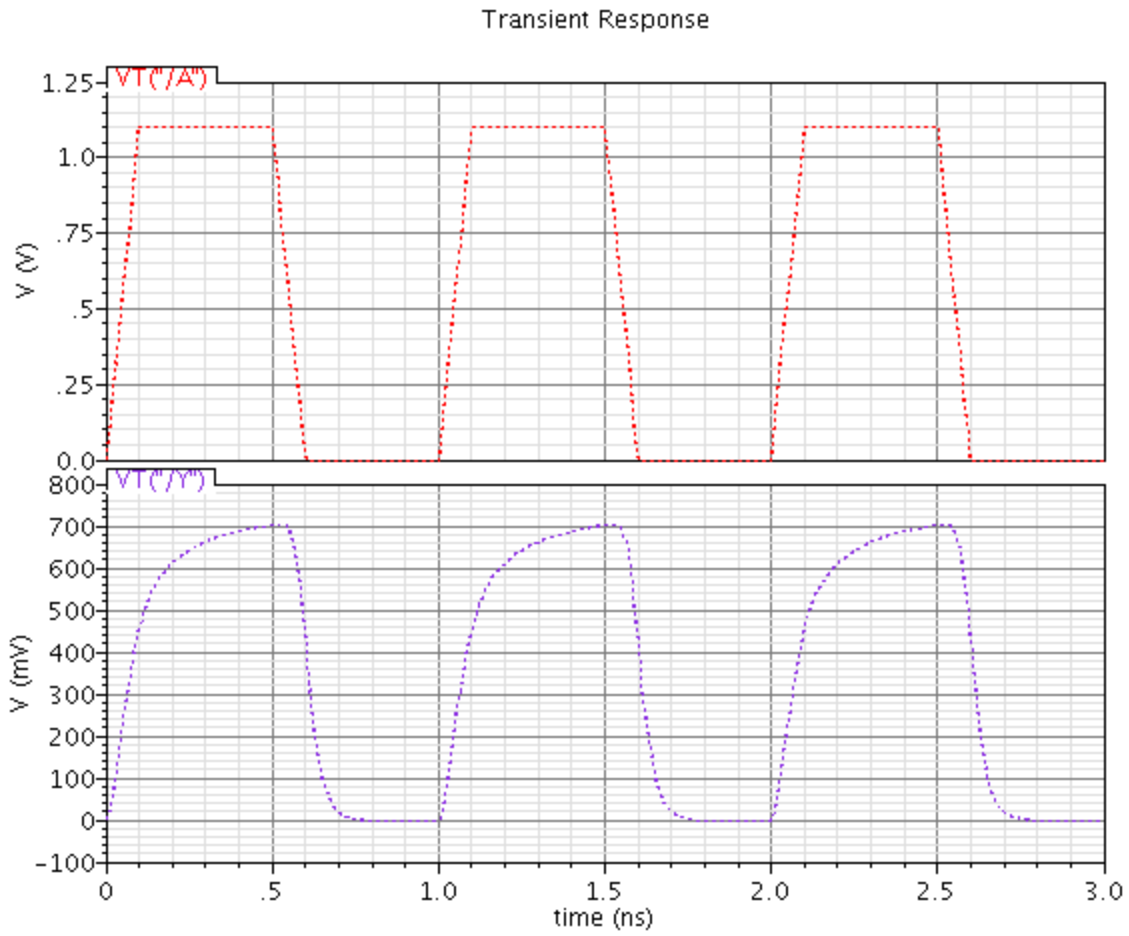


Figure 12: Simulations for the basic pass A component that we used to calculate the worst case delay and the scenario associated with it.

Register:

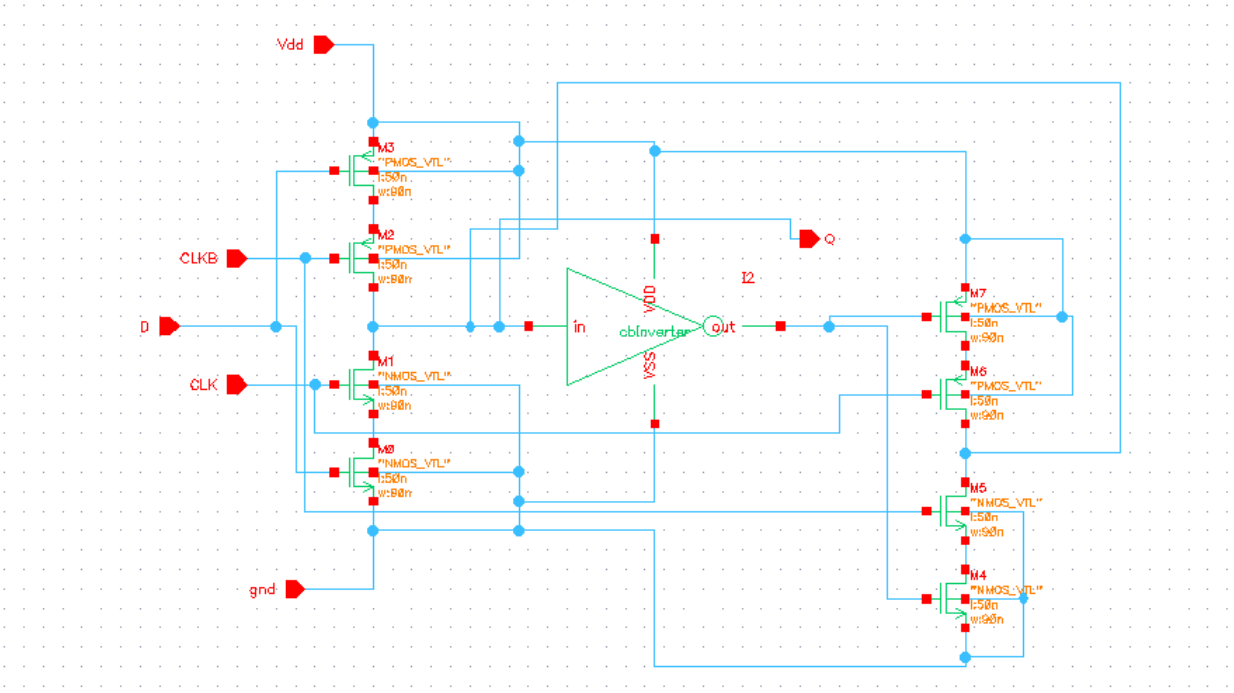


Figure 13: This is the base register for the master-slave register we used in previous homework as SE1.

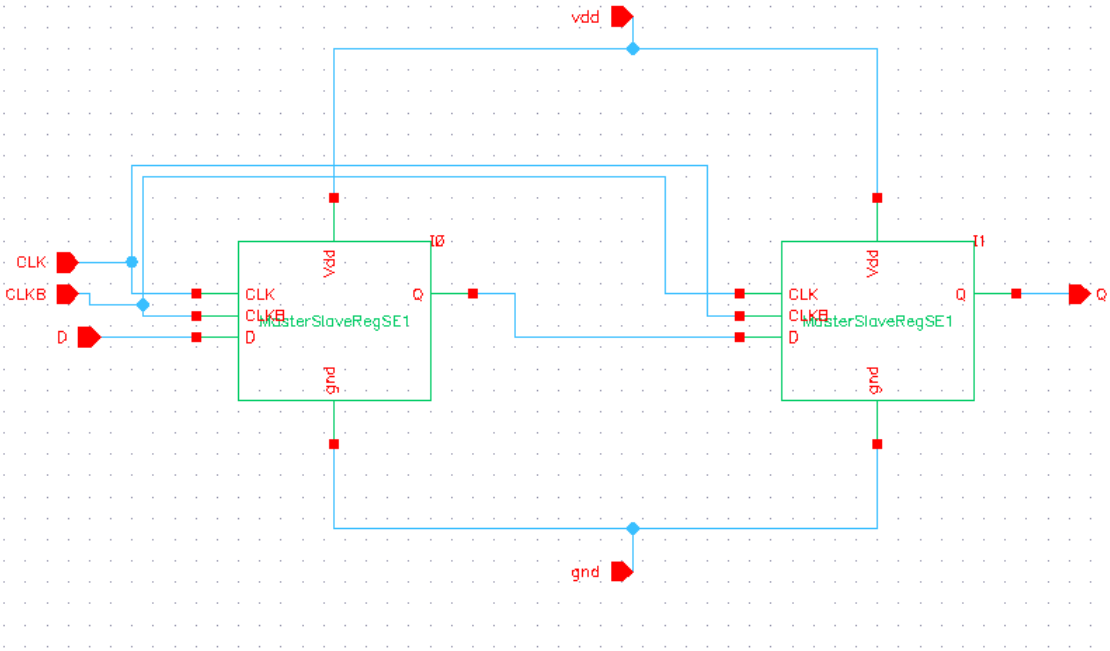


Figure 14: The schematic above is the master-slave register that was SE2 in the previous homework.

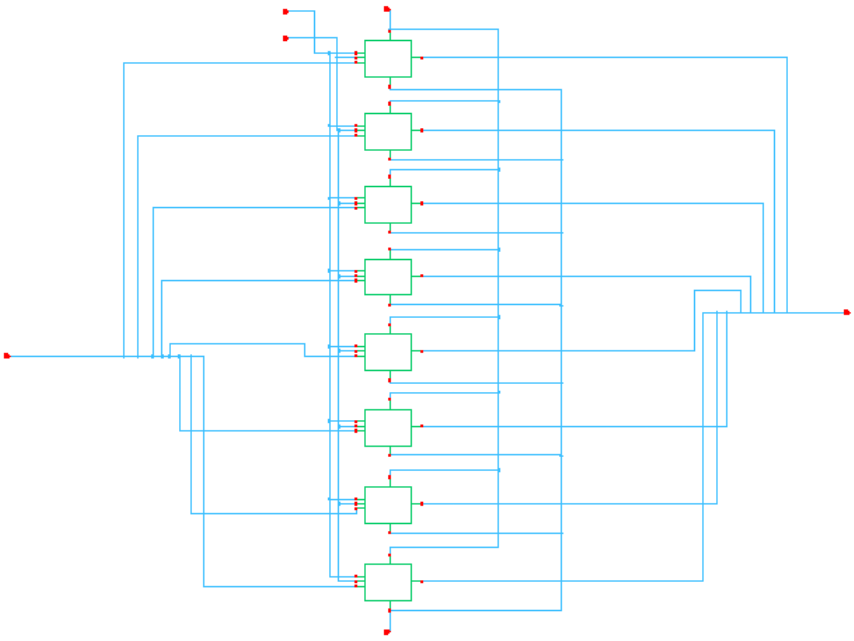


Figure 15: We used the master-slave registers we created to build an 8 bit register.

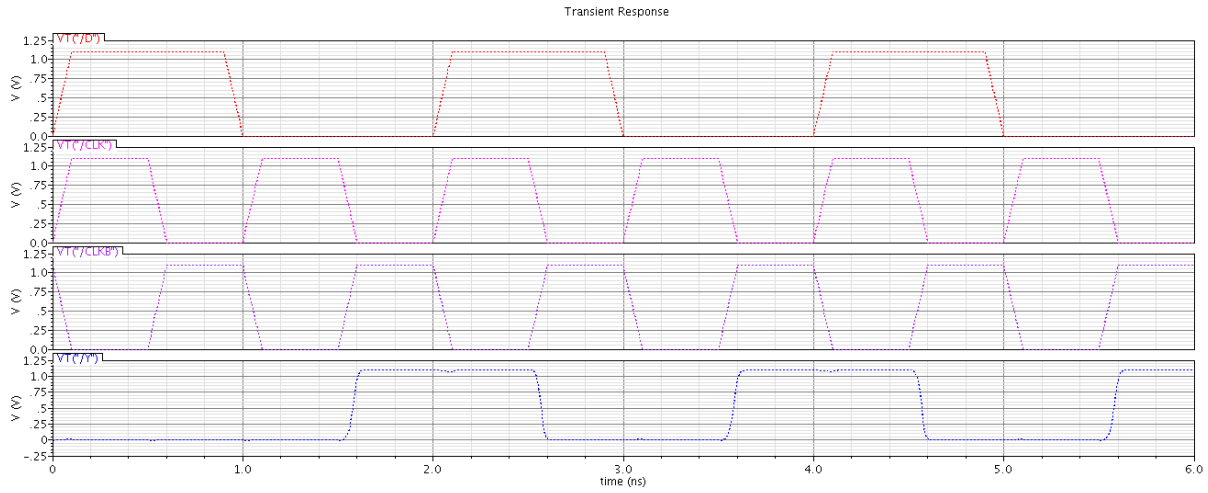


Figure 16: This the simulation for the master-slave register, SE2, that we used to get the worst-case delay and scenario associated with it.

XOR:

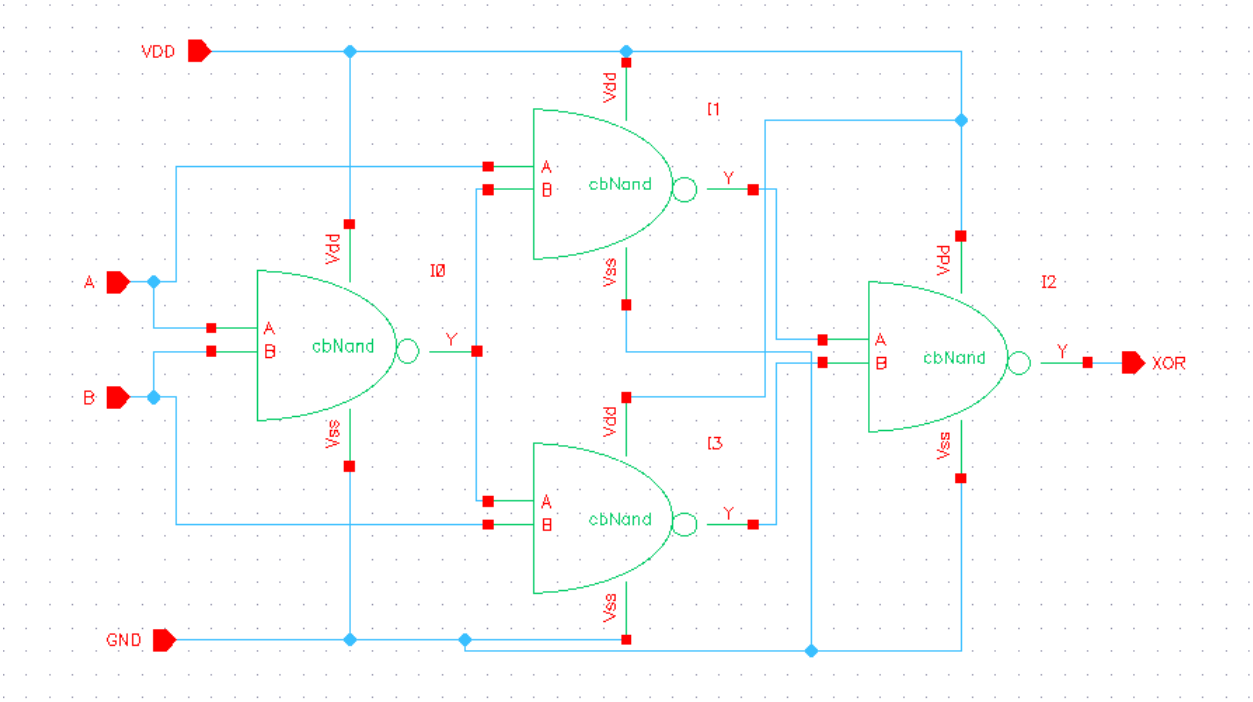


Figure 17: We created the XOR gate since this component was need to build the full adder that was used in the 8 bit adder and 8 bit two's component.

ALU:

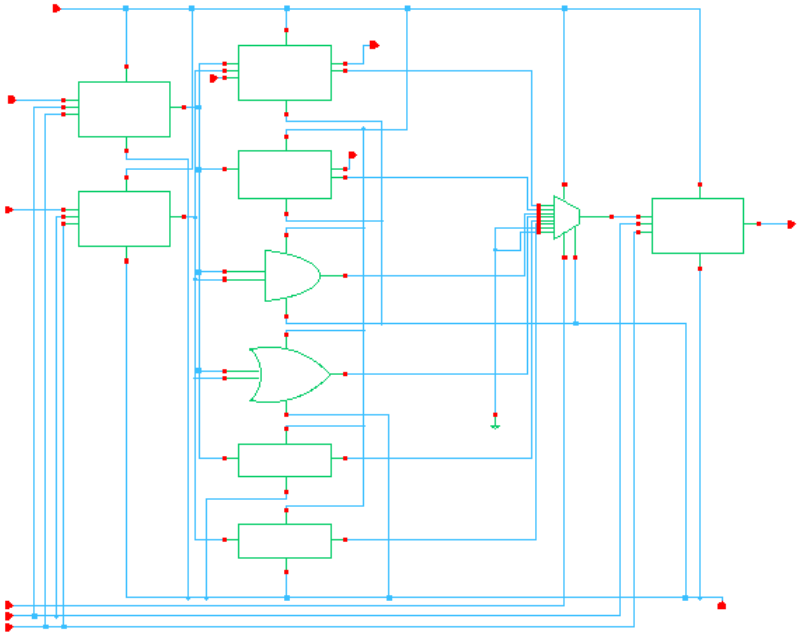


Figure 18: ALU we created in cadence that combines all the components we created.

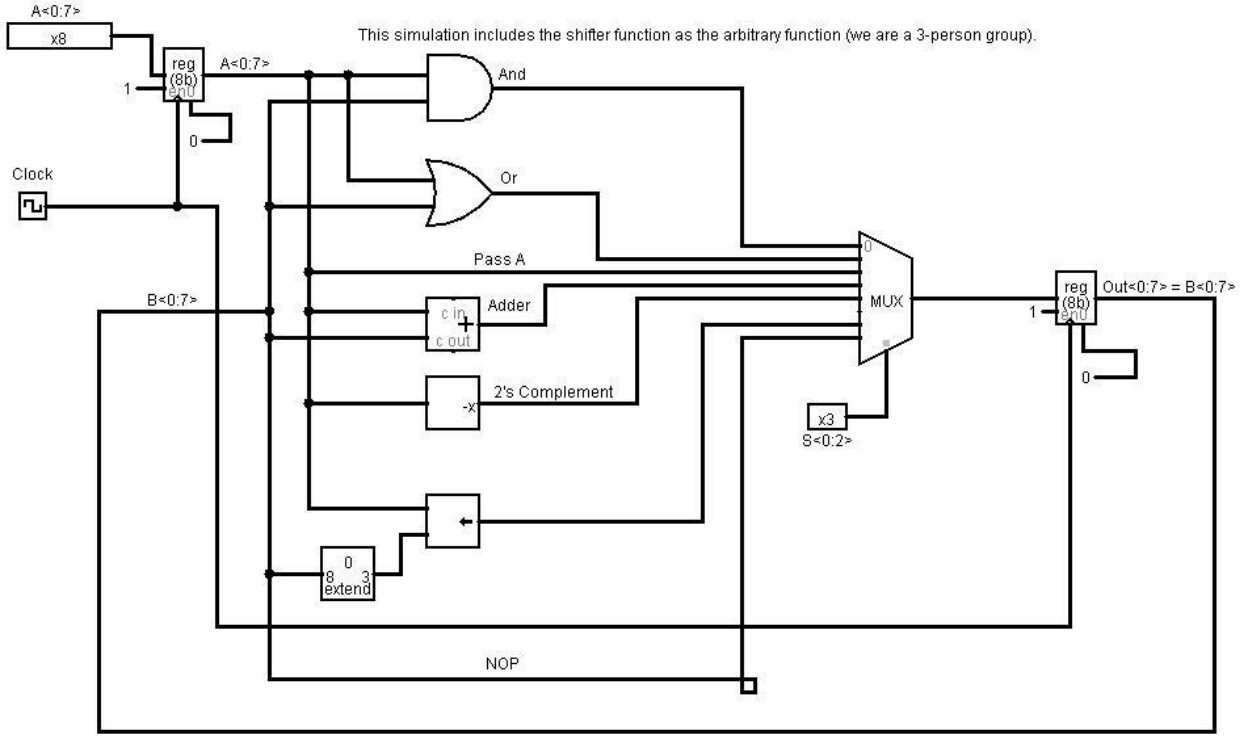


Figure 19: This is the final ALU that we will create in Logisim. This ALU schematic includes our arbitrary function of a shifter.

Shifter:

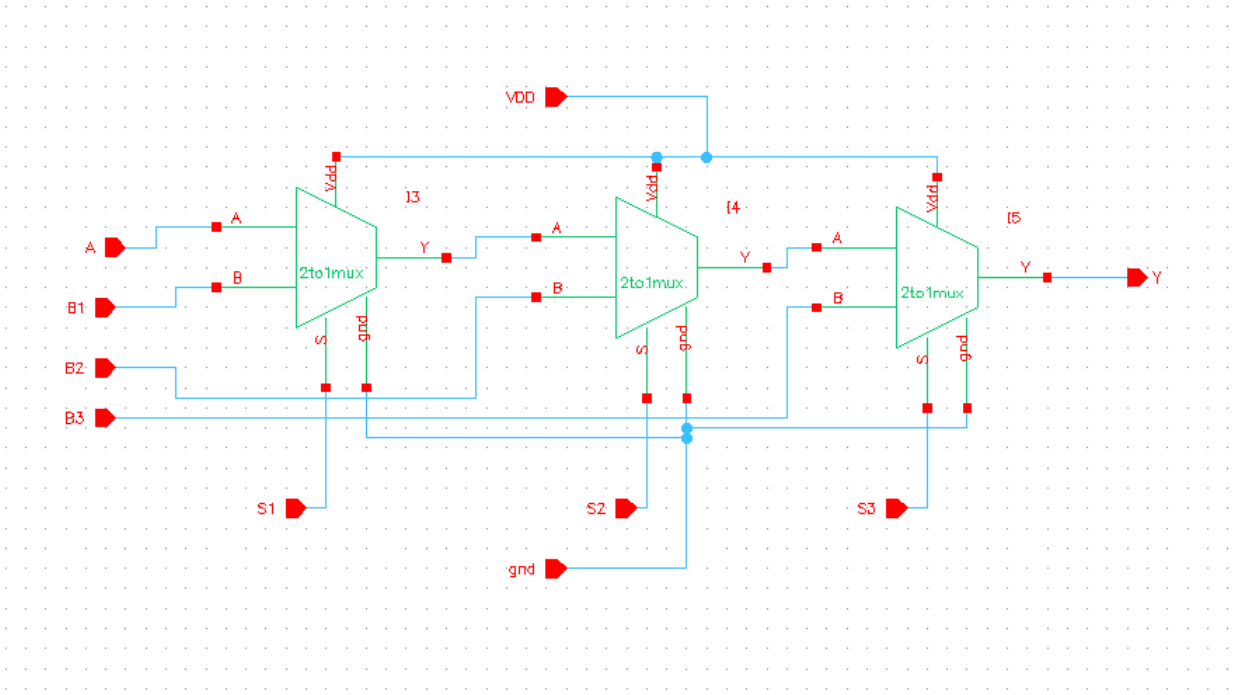


Figure 20: This is the 1 bit shifter we created for our arbitrary function in the ALU of an 8 bit shifter.

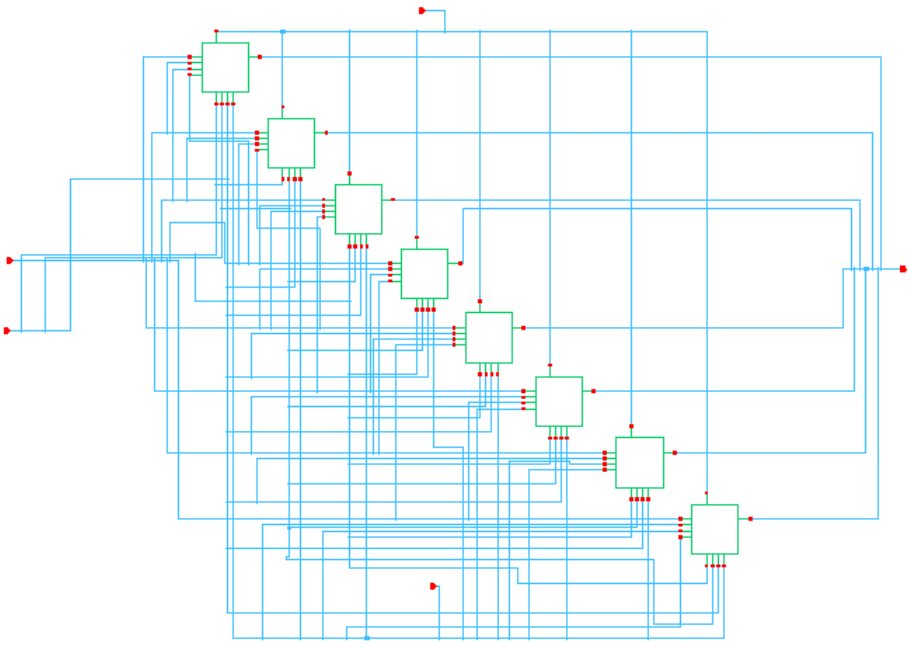


Figure 21: We used 8 shifters to create the 8 bit shifter we would use as our arbitrary function in the ALU.

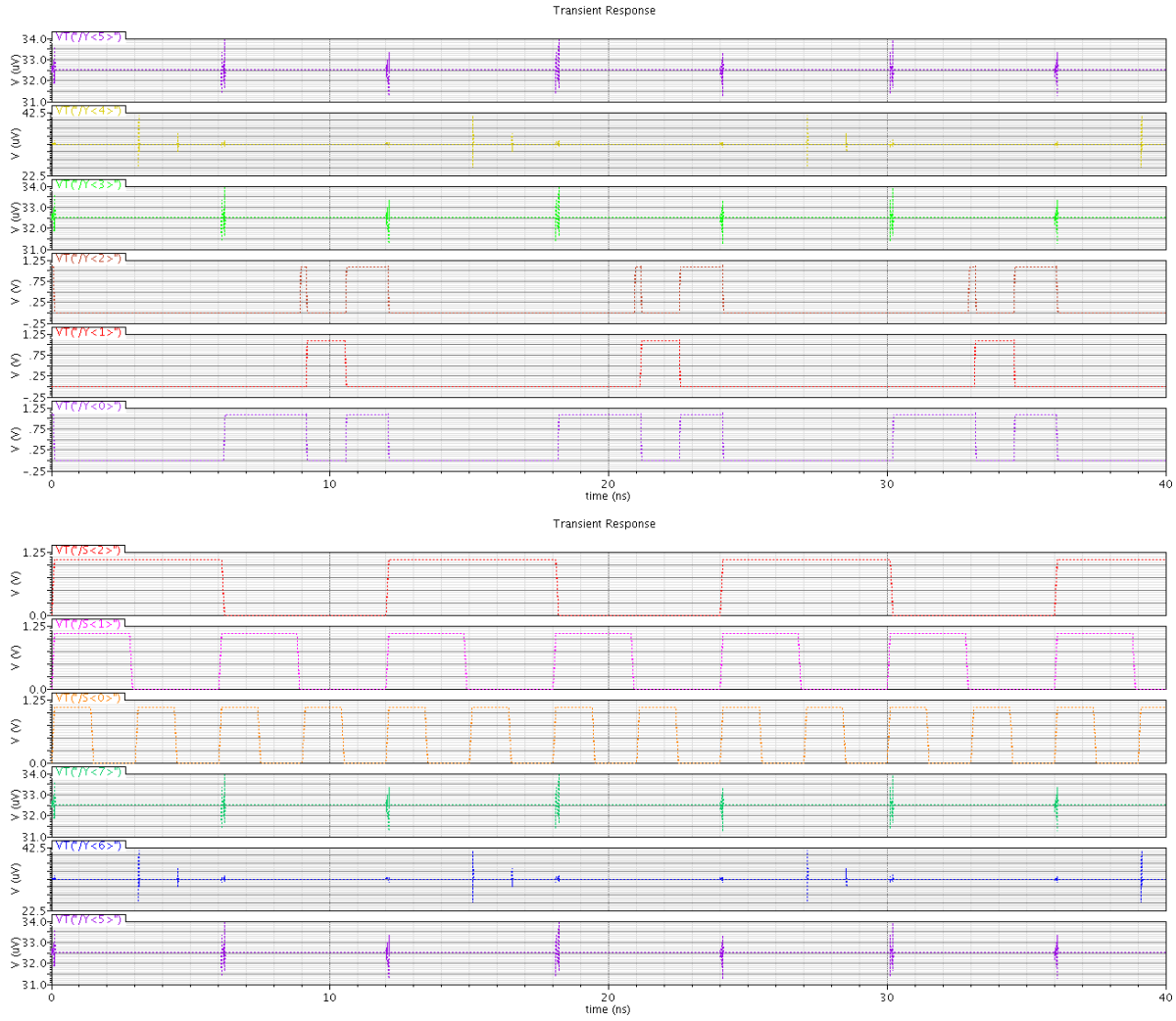


Figure 22, 23: These are the simulations we got to validate that the shifter works.

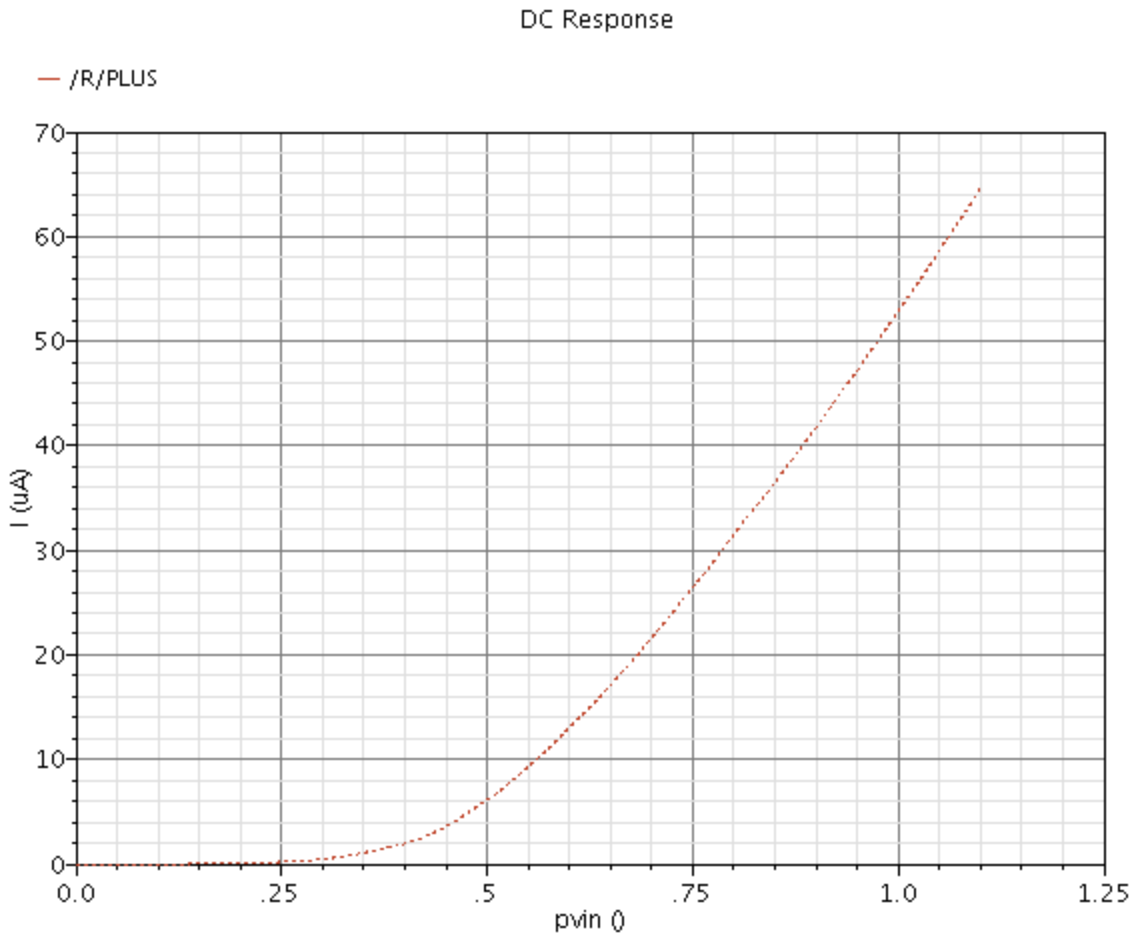


Figure 24: This is the graph we got for the current for the ALU function and which we calculated the energy using the resistor and the current flowing through the resistor.

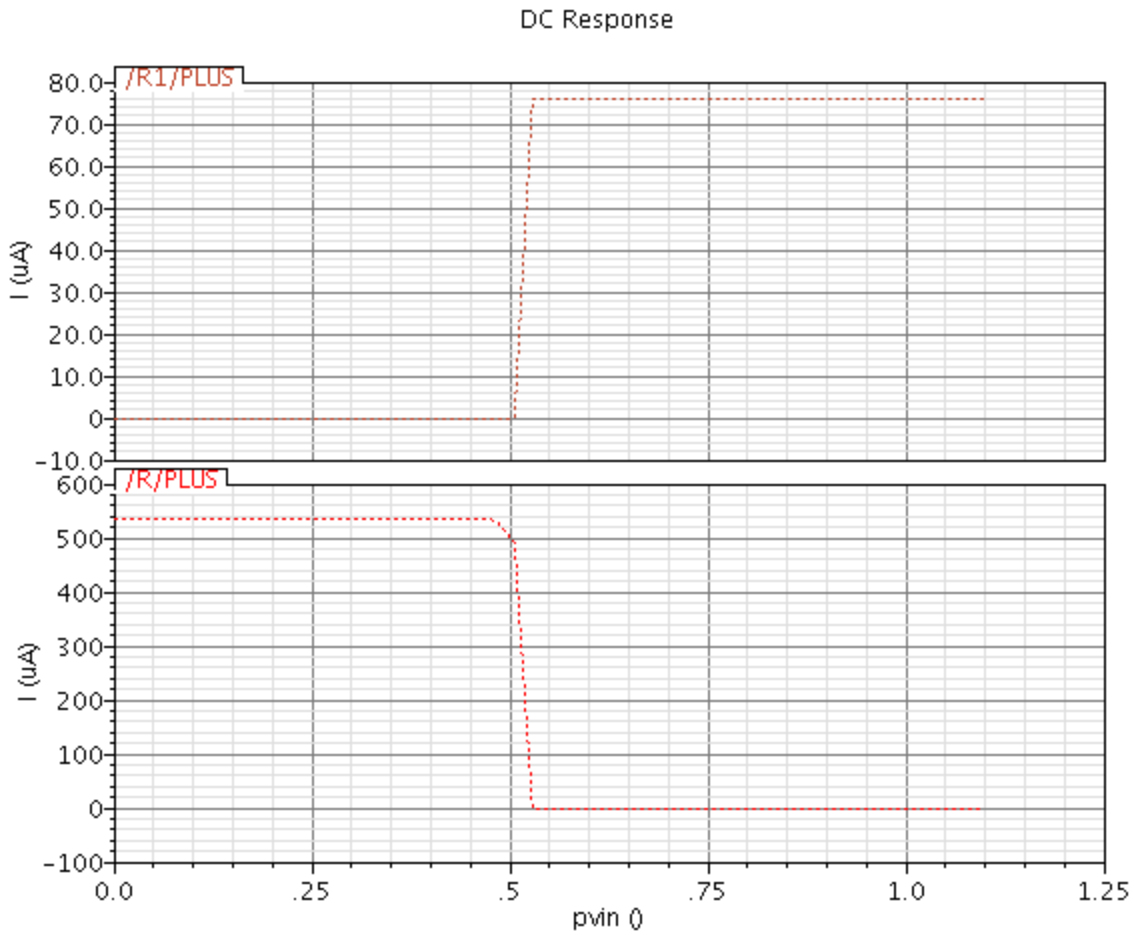


Figure 25: This is the graph we got for the current for the ADD function and which we calculated the energy using the resistor and the current flowing through the resistor.

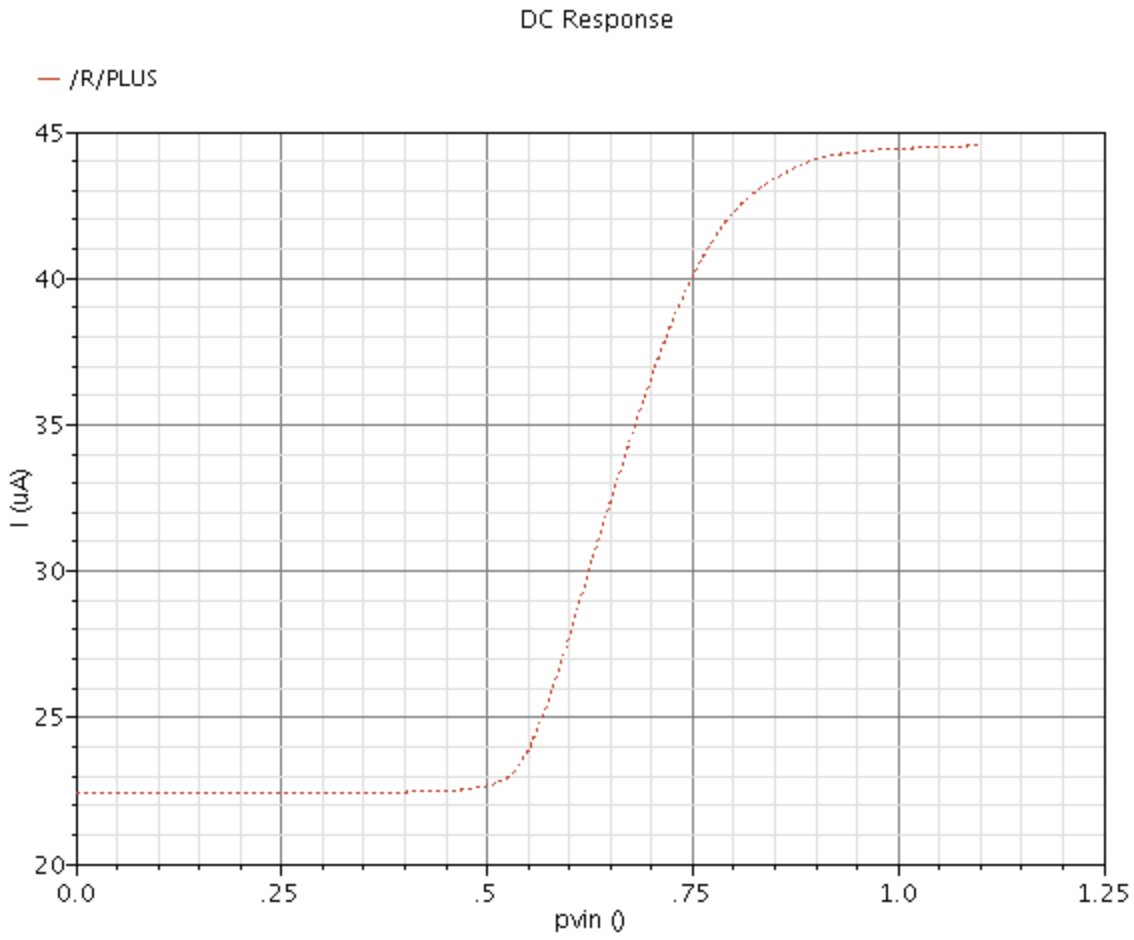


Figure 26: This is the graph we got for the current for the AND function and which we calculated the energy using the resistor and the current flowing through the resistor.

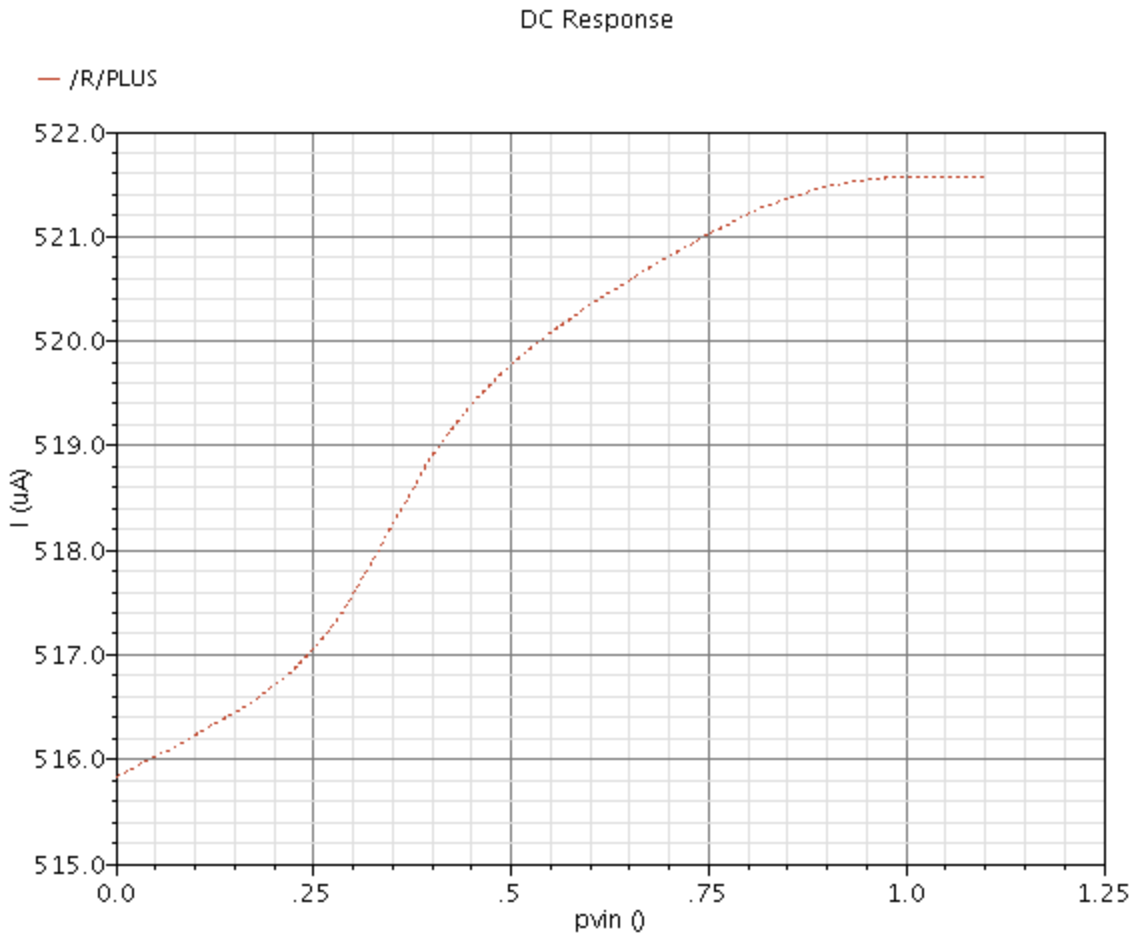


Figure 27: This is the graph we got for the current for the OR function and which we calculated the energy using the resistor and the current flowing through the resistor.

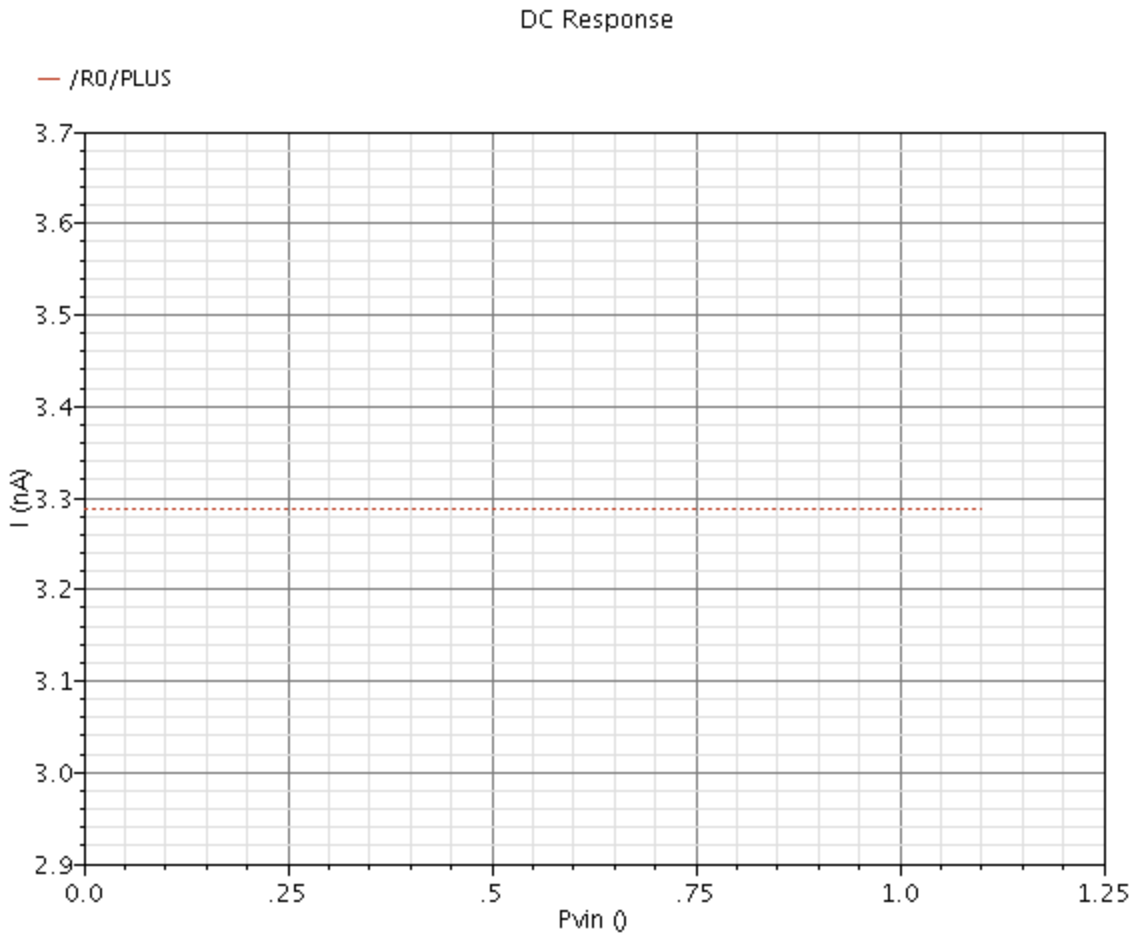


Figure 28: This is the graph we got for the current for the PASS A function and which we calculated the energy using the resistor and the current flowing through the resistor.

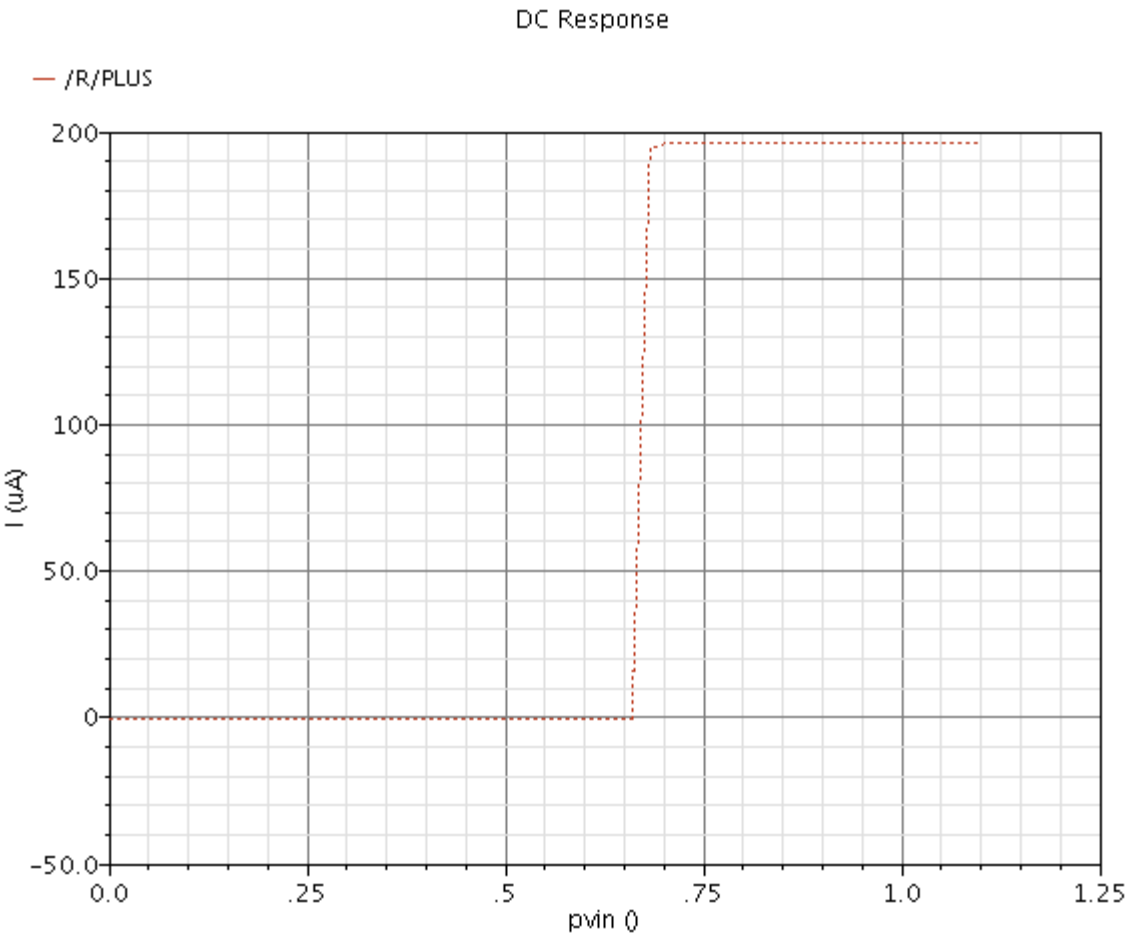


Figure 29: This is the graph we got for the current for the Shifter function and which we calculated the energy using the resistor and the current flowing through the resistor.